

|  |  |
| --- | --- |
|  | **SOFTWARE REQUIREMENT SPECIFICATION** |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |
| --- |
|  |

[APPOINTMENT BOOKING SYSTEM]

|  |  |  |
| --- | --- | --- |
| VERSION: [2] | REVISION DATE: [29/03/15] |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Approver Name** | **Title** | **Signature** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SCHOOL OF COMPUTER ENGINEERING**

NANYANG TECHNOLOGICAL UNIVERSITY

**Contents**

[APPOINTMENT BOOKING SYSTEM]………………………………………………………...i

*1. Product Description* 1

1.1 Purpose 1

1.2 Scope 1

1.3 Stakeholders and Users 2

1.4 Assumptions 2

1.5 Constraints 3

*2. Functional Requirements* 3

2.1 ABS must allow Patient to register an account for the application 3

2.2 ABS must allow Patients to log into their account 3

1. ABS must allow Patients to make changes to their password, address, email, contact number and preferred mode of communication. 4

2. ABS must pass Patients’ details to server to update the changes. 4

2.4 ABS must allow Patients to create their appointments 4

2.5 ABS must allow Patients to edit their appointments 4

2.6 ABS must allow Patients to delete their appointment 4

2.7 ABS must display Patient’s outstanding appointment details 5

*3. Non-Functional Requirements* 6

3.1 Usability 6

3.2 Reliability 6

3.3 Supportability 6

*4. Interface Requirements* 7

4.1 User Interfaces 7

4.2 Hardware Interfaces 10

4.3 Software Interfaces 10

*5.* *Use Case Model* 11

5.1 Use Case Diagram 11

5.2 Use Case Description. 12

*6.* *Data Dictionary* 25

*7.* *References* 25

*8.* *Revision History* 25

## *1. Product Description*

### 1.1 Purpose

(Provide an overview of the business organization sponsoring the development of the software application, including the mission statement and organizational objectives of the business unit)

The “Appointment Booking System” is developed by a group of 5 IT professionals, from AceCoders Pte Ltd. The purpose of this document is to give a detailed description of the requirements for the “Appointment Booking System” (ABS) mobile application. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to our client, Green Health Clinic Pte Ltd, for their consideration and a reference for developing the future upgrades of the system for the client. The mission statement for AceCoders is to develop functional and reliable applications that comply with the existing software engineering protocols. The organizational objectives are to ensure that client’s requirements are satisfied and provide user-friendly application.

### 1.2 Scope

(Describe the scope of the software application to be produced)

The “Appointment Booking System” is a mobile application which allows patients to manage their appointments. The software will be implemented on Android platform at Green Health Clinic in its initial stages.

Patient can only register an account using his/her NRIC number. Information such as name, date of birth and contact numbers etc. are required to be filled up during registration. Patient can login and use the ABS system after successfully creating an account.

Patient can create medical appointment with consultant through ABS system. Patient can also view, change or cancel his/her appointment after appointment is made. These appointment records will be stored inside the database managed by Green Health Clinic.

### 1.3 Stakeholders and Users

(Identify the stakeholders.

Identify each type of user of the software by function, location, and type of device. Specify the nature of their use of the software application.)

Table : Stakeholders and Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stakeholders** | **Function** | **Location** | **Type of Device** | **Nature of Use** |
| Patient | To view,create, update or cancel medical appointments using ABS system | Anywhere with an Internet connection | Android mobile devices with ABS application installed | Managing medical appointments |
| Administrator of the software application | To check, manage, create upgrades for system | Admin office | Computer with capabilities to access the system’s backend | Managing of overall system |

### 1.4 Assumptions

(Describe the assumptions that can affect the requirements specified in this SRS.)

1. Patients have installed the ABS application on their Android phone.
2. There will not be any connection disruptions throughout the appointment making process.
3. Patients uses only their own personal details (NRIC, name, etc.) for the usage of ABS application.
4. The clock on the device is accurate and correct (Singapore’s local time) to allow function synchronization.
5. Patients will not create appointments that they do not require..
6. Patients’ account can only be logged into 1 device at a time.
7. Green Health Clinic’s central database server will not be down when the Patient is accessing the application.
8. By registering for the application, the Patient has agreed to abide to the client’s terms and conditions.
9. The display of ABS will not be affected by the Patients’ device resolution.
10. \*\*\*Try email function for forget password, if cannot use security question instead

### 1.5 Constraints

(Describe the constraints that can affect the requirements specified in this SRS.)

1. Internet connection is a constraint for ABS. Since ABS fetches data from the database over the Internet, it is essential that patients and the server have a working Internet connection in order for ABS to function.
2. Capacity of the database is another constraint. Since the database stores all the patients’ information, the amount of data might be too overwhelming for the database system.
3. Each patient can only have up to a maximum of one outstanding appointment with each of the specialized clinics.
4. Database connectivity is a constraint as the ABS application depends on the database to verify the Patient’s information before allowing them to access their personal records.

## *Functional Requirements*

State functional requirements in a verifiable and atomised fashion. (describe interactions between the system and the environment, to map program inputs to program outputs. E.g information to be displayed, or interface with other systems.+ USE CASE MODEL AS TOOL FOR FUNC REQ)

### ABS must allow Patient to register an account for the application

* + 1. ABS must request Patients to fill in their NRIC, name, password, gender, date of birth, address, phone number, email and preferred mode of communication.
    2. ABS must verify that the username is unique.
    3. ABS must verify that their password is 8 to 12 character long.

### ABS must allow Patients to log into their account

* + 1. ABS must prompt user for username and password.
    2. Patients’ username is their NRIC number.
    3. ABS must pass the username and password keyed in by Patients to the database server for verification.
    4. ABS must direct user to the homepage only upon successful login.
    5. ABS must inform user upon unsuccessful login and remain on the login page.
  1. **ABS must allow Patients to update their profile**
     1. ABS must allow Patients to make changes to their password, address, email, contact number and preferred mode of communication.
     2. ABS must pass Patients’ details to server to update the changes.

### ABS must allow Patients to create their appointments

* + 1. ABS must allow Patients to choose the specialist clinic they need.
    2. ABS must only allow Patients to create appointments that are at least 2 weeks in advance
    3. ABS must only allow Patients to select the timing of the appointments from a list of fixed time slots of 1 hour each.
    4. ABS must allow Patient to select the type of appointment that are going for, i.e. consultation, blood test, etc.
    5. Patients must fill in all mandatory fields that includes date, time, type of appointment and clinic
    6. ABS must provide an optional field for Patient to describe symptoms they are experiencing.
    7. ABS must not allow any male Patient to create an appointment at the Women Health’s clinic.
    8. ABS must pass the data to server to verify that Patients does not have any outstanding appointment at the same clinic.
    9. ABS must pass the data to server to verify the appointment for the time and date is not already created by another Patient.
    10. ABS must pass the data to server to verify that user satisfy the pre-requirements tests according to the type of appointment booked before the appointment can be created.

### ABS must allow Patients to edit their appointments

* + 1. ABS must allow Patients to change the time, date and type of appointment.
    2. ABS must only allow Patients to edit appointments at most 1 week in advance.
    3. ABS must only allow Patients to select the timing of the appointments from a list of fixed time slots of 1 hour each.
    4. Patients must fill in all mandatory fields that includes date, time and type of appointment.
    5. ABS must provide an optional field for Patient to describe symptoms they are experiencing.
    6. \*\*\*ABS must be able to pass the data to server to verify the appointment for the time and date is not already created by another Patient.

### ABS must allow Patients to delete their appointment

* + 1. ABS must allow Patients to select the appointment that they are going to delete.
    2. ABS must pass appointment data to server to process deletion.

### ABS must display Patient’s outstanding appointment details

* + 1. ABS must allow Patients to view their appointments from the 3 specialist clinics.
    2. ABS must display Patients’ appointment details including, time, date, clinic, type of appointment, description of symptoms and pre-assessment test.
  1. **ABS must allow Patients to log out of out of their account**
     1. ABS must inform Patients upon successful logout and direct Patients to the login page.
     2. ABS must clear Patients’ login session when they log out.

## *Non-Functional Requirements*

State non-functional requirements in a verifiable and atomised fashion. (describe the properties the system must have, that is not directly related to the functional behaviour of the system.)

### Usability

* + 1. Speed of Use
       1. The system shall connect to the database and display the necessary information within 20 seconds.
       2. The system must restore session within 30 seconds after network error.

### Reliability

* + 1. Security
       1. \*\*\*The system must establish secure connection with the database server.
       2. The system shall allow users to be idle at the main page for a maximum of 10 minutes.
       3. \*\*\*The system must prompt Patients to re-enter their username and password after network is restored.

### Supportability

* + 1. Ease of Installation
       1. Patients can install the software on their Android mobile phones or devices through an installation file easily.
    2. Compatibility
       1. ABS must be compatible with the Android system.
       2. \*\*\*ABS must be able to dynamically adjust it UI layout and resolutions for different Android mobile devices.

## *4. Interface Requirements*

### 4.1 User Interfaces

Describe the screen design requirements. Include mock-ups and any report format.

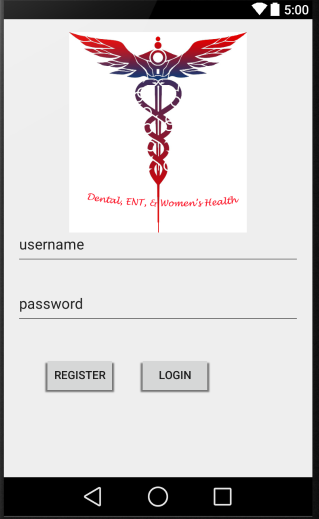
Users should see the Welcome page screen when first open the ABS app. Users can choose to login. First time users can choose to register.

Figure : UIWelcome

First time users who register will be directed to our Registration Page. By clicking “Next”,

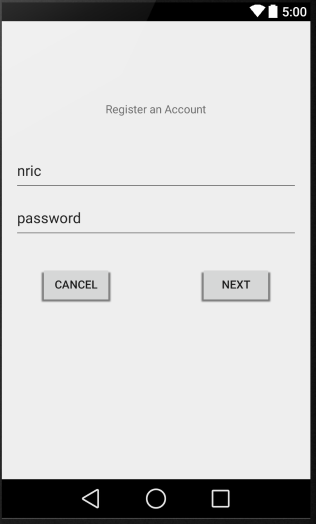
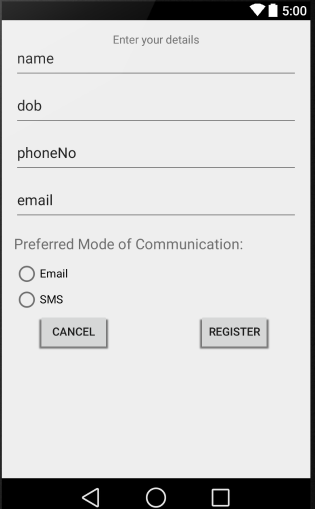
 Users can register by filling up the required fields as seen in Figure 3.

Figure 3: UIRegistration2

Figure 2: UIRegistration1

// Update new screenshot when everything is done

Users must click the “Register” button to submit the registration form. The app should return to Welcome page upon successful registration.

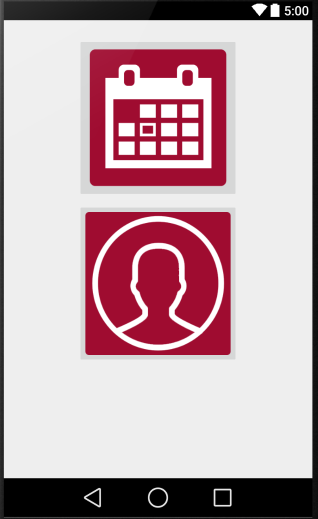
After User logs in to account, they will be directed to the Home page (Appointment Displays page? Menu at the side), where users can choose to manage appointments or update their profile as seen in Figure 4.

Figure 4: UIHome

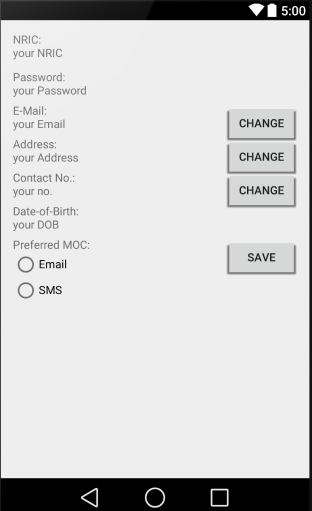
At the manage profile interface, Users are allowed to change their email, address, contact number and preferred mode of communication. The changes will be saved upon clicking the “Save” button. How bout change password?

Figure 5: UIManageProfile

At the manage appointment interface, Users can choose to display appointments which shows users a record of current and past appointments whereby users can then change, or cancel the appointment. Users can also choose to make new appointment at the manage appointment page as seen in Figure 6 below.

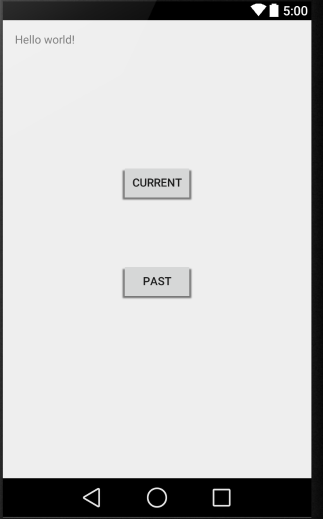
The Display Appointment page should display two options, the Current and Past appointment as seen in Figure 7. At the Current Appointment page, users can choose to change or cancel their current appointment. (Picture?) The Change Appointment page should display the details of the appointments that are already made, such as date and time, for users to make necessary changes. The Cancel Appointment page should display the list of appointments the users have, with details such as the name, date and time, clinic and remarks. Users can select the appointment they wish to cancel by clicking the cancel option. While at the Past page, users can see a record of the past appointments they turned up for, which include details such as the date, time, clinic, doctor, and remarks. (Have?)

Figure 7: UIDisplayAppointment

Figure 6: UIManageAppointment

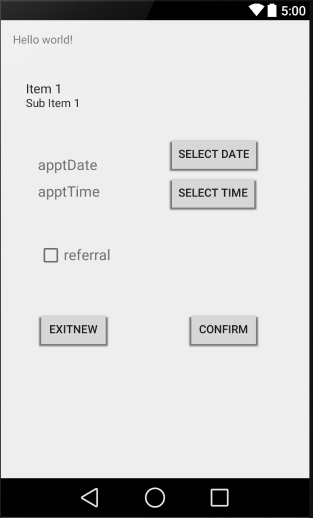
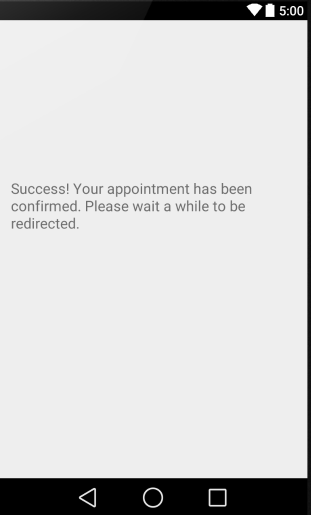
At the Create New Appointment page as seen in Figure 8, users can choose the date and time of their desired appointment. Users should see a calendar of dates for them to select from. Users can only book appointments two weeks in advance. Upon selection of date, users should see a clock (Removed clock. Drop down list?) for them to select the time of the appointment. User should see a confirmation page for their new appointment, as seen in Figure 9.

Figure 9: UIConfirmAppointment

Figure 8: UINewAppointment

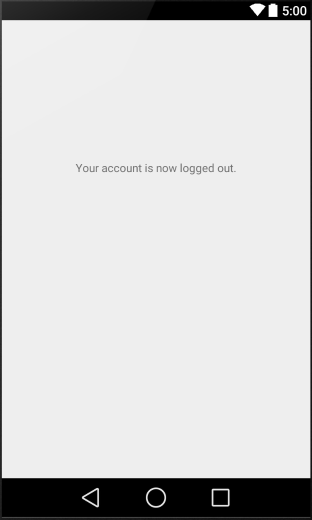
Finally, when users log out of the ABS application, there will be a logout confirmation page as shown in Figure 10 below. Users will be directed to Welcome page.

Figure 10: UILogout

### 4.2 Hardware Interfaces

Describe how the software application interfaces with hardware that exists outside the scope of the system.

The designated hardware of ABS would be an Android device, specifically an android mobile phone. The software also make use of the clock setting of the device in its function. The hardware connection to the database server is managed by the Android operating system and the Green Health Clinic’s server to update the patient’s personal information and appointment details.

### 4.3 Software Interfaces

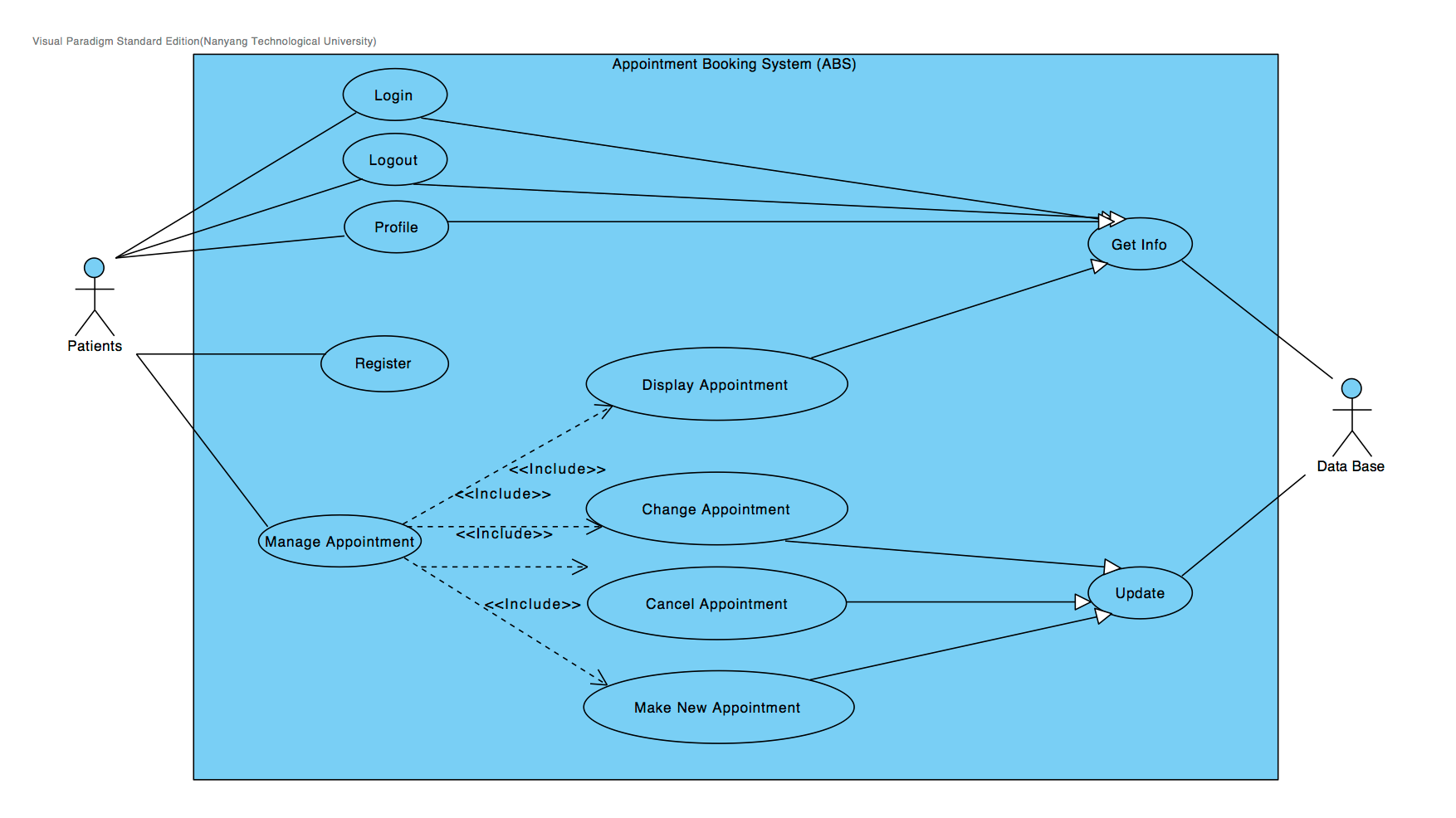
Describe how the software application interfaces with software systems that exist outside the scope of the system.

ABS communicates with the Green Health Clinic’s database server to retrieve details about the Patient and their appointments. The communication between ABS and Green Health Clinic’s database server only includes operation that reads the data and send any other relevant information to the server. The server will be in charge of any other modifications and is not a responsibility of ABS application.

## *5. Use Case Model*

Provide the top-level use case diagram, followed by the use case description for each use case.

### 5.1 Use Case Diagram



### 5.2 Use Case Description.

**1.**   **Register**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Register | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | Patient | | |
| Description: | This use case describes how patients can make registration via the Appointment Booking System (ABS). | | |
| Preconditions: | 1.      1. Patient has installed ABS app onto phone.  2.      2. Patient is not logged in.  3.      3. ABS app is opened.          4. Patient does not have an ABS account. | | |
| Postconditions: | On success:   1. Patient created registration and logged in.   On failure:   1. User is unable to register to app. | | |
| Priority: | High. | | |
| Frequency of Use: | Always. | | |
| Flow of Events: | 1. ABS prompts for new registration. 2. Patient enters NRIC as username. 3. Patient enters 8 to 12 characters password, including both upper- and lower-case letters, numerical digits and special characters. 4. Patient re-enter password. 5. Patient enters real name. 6. Patient enters date of birth. 7. Patient enters race. 8. Patient enters contact number. 9. Patient enters Email address. 10. Patient enters mailing address. 11. Patient choose preferable mode of communication, checkbox for SMS and Email. 12. Patient presses ‘Confirm ’ button submits the application form. 13. ABS check if NRIC is valid. 14. ABS check if password matches the conditions imposed. 15. ABS check if both passwords entered is the same. 16. ABS create account for patient and saved the information entered into ABS database 17. ABS directs to ManagementAppointment page upon successful registration. | | |
| Alternative Flows: | 16.  16.1 Patient registration is rejected when required fields are either not filled in or filled in improperly.  16.2 ABS returns to registration page with empty necessary fields. | | |
| Exceptions: | Patient selects cancel option, use case action terminates. | | |
| Includes: |  | | |
| Assumptions: | 1. Patient register with their own NRIC number. 2. Patient that have a medical record with the medical practice will have already have their account created with a pre-allocated password. 3. New Patient without an account are not patients of the medical practice yet. | | |

### 

**4.**     **Manage Appointments**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | ManageAppointments | | |
| Created By: |  | Last Updated By: |  | |
| Date Created: |  | Date Last Updated: |  | |
| Actor: | Patients | | |
| Description: | This use case describes how a patient can access the 4 different functions (DisplayAppointment, CreateNewAppointment, ChangeAppointment and CancelAppointment). | | |
| Preconditions: | 1.  1. Patient has installed ABS app onto phone.      2. ABS app is opened.      3. Patient has an ABS account.      3. Patient is logged in.      4. Patient’s ManageAppointments session has not expired. | | |
| Postconditions: | 1.  Patients will be directed to DisplayAppointment, CreateNewAppointment, ChangeAppointment or CancelAppointment. | | |
| Priority: |  | | |
| Frequency of Use: | High. | | |
| Flow of Events: | 1. Patient clicks DisplayAppointment. 2. Proceed to Step 4.1. 3. Patient clicks CreateNewAppointment. 4. Proceed to Step 4.2. 5. Patient clicks ChangeAppointment. 6. Proceed to Step 4.3. 7. Patient clicks CancelAppointment. 8. Proceed to Step 4.4. | | |
| Alternative Flows: | 1. Patient clicks Cancel button  2. Use case action is terminated  3. Patient return to ManageAppointments.. | | |
| Exceptions: |  | | |
| Extends: | DisplayAppointment, CreateNewAppointment, ChangeAppointment, CancelAppointment | | |
| Assumptions: | 1. | | |

**4.1.**      **Display Appointments**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4.1 | |
| Use Case Name: | DisplayAppointments | |
| Created By: |  | Last Updated By: |
| Date Created: |  | Date Last Updated: |
| Actor: | Patient | |
| Description: | This use case describes how a patient can choose to view his/her current/previous appointments. | |
| Preconditions: | 1.  Patient is logged in.      Patient is currently in HomePage.       Patient selects ManageAppointments. | |
| Postconditions: | On success:  1.    Patient is shown the appointments that he/she desires.  On failure: | |
| Priority: |  | |
| Frequency of Use: | Always. | |
| Flow of Events: | 1.      1. Patient clicks DisplayAppointments button.  2.      2. ABS displays a new screen which shows “Current Appointments” and “Appointment History”  3.      3. Patient selects to view either. | |
| Alternative Flows: | 1. Patient clicks cancel button. 2. Use case action is terminated. 3. Patient returns to previous screen that they are at. | |
| Exceptions: |  | |
| Includes: | GetInfo | |
| Assumptions: |  | |

**4.2.**     **Make New**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.2 | | |
| Use Case Name: | Make New Appointment | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | Patient | | |
| Description: | This use case describes how a Patient adds a new appointment . | | |
| Preconditions: | Patient is already logged in. | | |
| Postconditions: | On success:   1. Patient’s appointment information is sent to **Update** use case. 2. Patient is directed back to appointment page.   On failure:   1. ABS system will direct user back to same page. | | |
| Priority: |  | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | 1. ABS will display making new appointment sequence. 2. Patient selects clinic to go to. 3. Patient select the date for the appointment for which he/she wants 4. ABS app will get information for available time slots from **Get Info** use case and is displayed to user. 5. Patient select the timeslot 6. Patient key in the description of ailments 7. Patient chooses checkbox for either referral or repeat follow-up consultation 8. Patient chooses “repeat follow-up consultation” checkbox and confirm appointment. 9. ABS direct Patient back to Manage Appointment page. | | |
| Alternative Flows: | 8.       8.1 Patient chooses “referral” checkbox.       8.2 Patient key in information from referral doctor or clinic.       8.3 Patient confirm appointment.       8.4 ABS direct Patient back to Manage Appointment page. | | |
| Exceptions: | Patient selects cancel option, use case action terminates. | | |
| Includes: | GetInfo, Update | | |
| Assumptions: | 1. There is no collision of different Patients selecting the same time slot at the same time. | | |

**4.3.**      **Change Appointments**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.3 | | |
| Use Case Name: | ChangeAppointments | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | Patient | | |
| Description: | This use case describes how a patient can choose to change his/her current appointments. | | |
| Preconditions: | 1.  Patient is logged in.      Patient is currently in HomePage.       Patient selects ChangeAppointments. | | |
| Postconditions: | On success:  1.    1. Patient is shown all his/her current appointments.        2. Patient can select which current appointment he/she wants to change.  On failure:  1. There is no appointment made, nothing to be change.  2. Patient is returned to Manage Appointments page. | | |
| Priority: |  | | |
| Frequency of Use: | Always. | | |
| Flow of Events: | 1.      1. Patient clicks ChangeAppointments button.  2.      2. ABS displays a new screen which shows all current appointments.  3.      3. Patient selects an appointment to change.          4. Patient brought to a screen to modify details of the selected appointment. Details include: clinic, date, timeslot, description. | | |
| Alternative Flows: | 1. Patient does not have any appointments to change 2. Patient returns to the ManageAppointments page. | | |
| Exceptions: | 1. Patient selects Back, use case option terminates. | | |
| Includes: | GetInfo, Update | | |
| Assumptions: |  | | |

**4.4.     Cancel Appointments**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.4 | | |
| Use Case Name: | CancelAppointments | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | | Patient | |
| Description: | | This use case describes how a Patient cancel his/her appointments. | |
| Preconditions: | | 1. Patient is already logged in.  2. Patient selects ‘Cancel appointments’ | |
| Postconditions: | | On success:   1. Patient can select which appointment to cancel. 2. Patient is shown a successfully cancelled               appointments message. 3. Appointment is terminated.   On failure:   1. User is unable to cancel appointment. 2. ABS returns to Manage Appointments page. | |
| Priority: | |  | |
| Frequency of Use: | | Always. | |
| Flow of Events: | | 1. Patient clicks CancelAppointments button. 2. ABS shows the list of current appointments. 3. Patient selects which appointment to cancel 4. ABS displays a “Confirm to cancel selected appointments?” message. 5. Patient clicks ‘Yes’ button to confirm cancel. 6. ABS terminates Patient’s appointment. | |
| Alternative Flows: | | 1. Patient does not have any appointment to cancel. 2. ABS returns to ManageAppointments page. | |
| Exceptions: | | Patient selects cancel option, use case action terminates. | |
| Includes: | | GetInfo, Update | |
| Assumptions: | |  | |

**5.**      **ChangeUserInfo**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | ChangeUserInfo | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | Patient | | |
| Description: | This use case describes how a patient can edit his/her personal information. | | |
| Preconditions: | 1. Patient is logged in. 2. Patient selects ChangeUserInfo. | | |
| Postconditions: | 1. Patient is allowed to modify several fields of his/her personal information. | | |
| Priority: |  | | |
| Frequency of Use: | Always. | | |
| Flow of Events: | 1.    1. Patient clicks ChangeUserInfo button.        2. Patient is brought to a screen displaying his/her editable personal information.  3.    3. Patient selects a specific field to edit.        4. Patient edits the field.        5. Patient saves the changes.        6. Patient is brought back to the ChangeUserInfo screen displaying the personal information. | | |
| Alternative Flows: | 1. Patient clicks back button. 2. Use case action is terminated. 3. Patient returns to previous screen that they are at. | | |
| Exceptions: |  | | |
| Includes: |  | | |
| Assumptions: |  | | |

**6.**      **Send message by email**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6 | | |
| Use Case Name: | SendMessageEmail | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | Patient | | |
| Description: | This use case describes how a patient can receive his/her appointment reminder by email. | | |
| Preconditions: | 1.  1. Patient has chosen email as preferred mode of notification.      2. Patient’s email is provided. | | |
| Postconditions: | 1.   Email is sent out. | | |
| Priority: |  | | |
| Frequency of Use: | Always. | | |
| Flow of Events: | 1.      1. Messaging Services gets Patient’s from GetInfo       1. Email Messaging Service gets appointment details from GetInfo use case.  2.      2. Email Messaging Service displays name of patient, venue of clinic, date and time of appointment, contact number of the clinic and ways of cancel appointment (i.e. via ABS app)  3.      3. Sent out email notification to patient. | | |
| Alternative Flows: |  | | |
| Exceptions: |  | | |
| Includes: | Get Info | | |
| Assumptions: | 1. Patient’s inbox is not full. | | |

**7.**    **Send Message by SMS**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7 | | |
| Use Case Name: | SendMessageSMS | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | SMS Messaging Service | | |
| Description: | This use case describes how a SMS Messaging Services will send out messages containing appointment details to the patient. | | |
| Preconditions: | 1. Contact number of the patient is provided. 2. Patient selects SMS mode of notification. | | |
| Postconditions: | 1.   SMS is sent out. | | |
| Priority: |  | | |
| Frequency of Use: | Always. | | |
| Flow of Events: | 1.      1. SMS Messaging Service gets appointment details from GetInfo use case.  2.      2. SMS Messaging Service displays name of patient, venue of clinic, date and time of appointment, contact number of the clinic and ways of cancel appointment (i.e. via ABS app)  3.      3. Sent out SMS notification to patient. | | |
| Alternative Flows: |  | | |
| Exceptions: |  | | |
| Includes: | Get Info | | |
| Assumptions: | 1. The contact number provided by the patient is accurate. | | |

**9.**      **GetInfo**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 9 | | |
| Use Case Name: | GetInfo | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | Database, invoked by Patients or SMS Service or Email Service | | |
| Description: | This use case describes how the database provides information to the initiating actors. | | |
| Preconditions: | 1.  Database is serviceable.      Valid Phone number or Email address | | |
| Postconditions: | On success:  1.   1.  Patients gain access to current and past appointment bookings via several use cases.       2. SMS service gains access to recent appointments and schedules messaging service.       3. Email service gains access to recent appointments and schedules emailing service.  On failure: | | |
| Priority: |  | | |
| Frequency of Use: | Variable | | |
| Flow of Events: | 1.      1. Patient activates “DisplayAppointment”, “ChangeAppointment”, or “CancelAppointment” use cases.          2. ABS retrieves users’ appointment information from database.     1. SMS service activates “SendMessageBySMS” use case. 2. SMS service retrieves users’ appointment information from database. 3. SMS service fulfills “SendMessageBySMS” use case. 4. Email service activates “SendMessageByEmail” use case. 5. Email service retrieves users’ appointment information from database. 6. Email service fulfills “SendMessageByEmail” use case. | | |
| Alternative Flows: | 1. Patient clicks back button. 2. Use case action is terminated. 3. Patient returns to previous screen that they are at. | | |
| Exceptions: |  | | |
| Includes: |  | | |
| Assumptions: |  | | |

## *6. Data Dictionary*

## *7. References*

Provide a list of all documents and other sources of information referenced in the SRS and utilized in developing the SRS. Include for each the document number, title, date and author.

| **Document No.** | **Document Title** | **Date** | **Author** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## *8. Revision History*

Identify changes to the SRS.

| **Version** | **Date** | **Name** | **Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |